

Extended MKCOL for Web Distributed Authoring and Versioning (WebDAV)

Abstract

This specification extends the Web Distributed Authoring and Versioning (WebDAV) MKCOL (Make Collection) method to allow collections of arbitrary resourcetype to be created and to allow properties to be set at the same time.

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the “Internet Official Protocol Standards” (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright © 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>¹) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

¹ <http://trustee.ietf.org/license-info>

Table of Contents

1	Introduction	3
2	Conventions Used in This Document	4
3	WebDAV Extended MKCOL	5
3.1	Extended MKCOL Support	5
3.1.1	Example: Using OPTIONS for the Discovery of Support for Extended MKCOL	5
3.2	Status Codes	5
3.3	Additional Precondition for Extended MKCOL	6
3.4	Example: Successful Extended MKCOL Request	6
3.5	Example: Unsuccessful Extended MKCOL Request	6
4	Using Extended MKCOL as an Alternative for MKxxx Methods	8
4.1	MKCALENDAR Alternative	8
4.1.1	Example: Using MKCOL Instead of MKCALENDAR	8
5	XML Element Definitions	11
5.1	mkcol XML Element	11
5.2	mkcol-response XML Element	11
6	Security Considerations	12
7	Acknowledgments	13
8	Normative References	14
Author's Address		15

1. Introduction

WebDAV [RFC4918] defines the HTTP [RFC2616] method MKCOL. This method is used to create WebDAV collections on the server. However, several WebDAV-based specifications (e.g., CalDAV [RFC4791]) define "special" collections -- ones that are identified by additional values in the DAV:resourcetype property assigned to the collection resource or by other means. These "special" collections are created by new methods (e.g., MKCALENDAR). The addition of a new MKxxx method for each new "special" collection adds to server complexity and is detrimental to overall reliability due to the need to make sure intermediaries are aware of these methods.

This specification defines an extension to the WebDAV MKCOL method that adds a request body allowing a client to specify WebDAV properties to be set on the newly created collection or resource. In particular, the DAV:resourcetype property can be used to create a "special" collection; alternatively, other properties can be used to create a "special" resource. This avoids the need to invent new MKxxx methods.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document uses XML DTD fragments (Section 3.2 of [W3C.REC-xml-20081126]) as a purely notational convention. WebDAV request and response bodies cannot be validated by a DTD due to the specific extensibility rules defined in Section 17 of [RFC4918] and due to the fact that all XML elements defined by this specification use the XML namespace name "DAV:". In particular:

1. Element names use the "DAV:" namespace.
2. Element ordering is irrelevant unless explicitly stated.
3. Extension elements (elements not already defined as valid child elements) may be added anywhere, except when explicitly stated otherwise.
4. Extension attributes (attributes not already defined as valid for this element) may be added anywhere, except when explicitly stated otherwise.

When an XML element type in the "DAV:" namespace is referenced in this document outside of the context of an XML fragment, the string "DAV:" will be prefixed to the element type.

This document inherits, and sometimes extends, DTD productions from Section 14 of [RFC4918].

3. WebDAV Extended MKCOL

The WebDAV MKCOL request is extended to allow the inclusion of a request body. The request body is an XML document containing a single DAV:mkcol XML element as the root element. The Content-Type request header MUST be set appropriately for an XML body (e.g., set to "text/xml" or "application/xml"). XML-typed bodies for an MKCOL request that do not have DAV:mkcol as the root element are reserved for future usage.

One or more DAV:set XML elements may be included in the DAV:mkcol XML element to allow setting properties on the collection as it is created. In particular, to create a collection of a particular type, the DAV:resourcetype XML element MUST be included in a DAV:set XML element and MUST specify the expected resource type elements for the new resource, which MUST include the DAV:collection element that needs to be present for any WebDAV collection.

As per the PROPPATCH method (Section 9.2 of [RFC4918]), servers MUST process any DAV:set instructions in document order (an exception to the normal rule that ordering is irrelevant). If any one instruction fails to execute successfully, all instructions MUST fail (i.e., either all succeed or all fail). Thus, if any error occurs during processing, all executed instructions MUST be undone and a proper error result returned. Failure to set a property value on the collection MUST result in a failure of the overall MKCOL request -- i.e., the collection is not created.

The response to an extended MKCOL request MUST be an XML document containing a single DAV:mkcol-response XML element, which MUST contain DAV:propstat XML elements with the status of each property when the request fails due to a failure to set one or more of the properties specified in the request body. The server MAY return a response body in the case where the request is successful, indicating success for setting each property specified in the request body. When an empty response body is returned with a success request status code, the client can assume that all properties were set.

In all other respects, the behavior of the extended MKCOL request follows that of the standard MKCOL request.

3.1 Extended MKCOL Support

A server supporting the features described in this document MUST include "extended-mkcol" as a field in the DAV response header from an OPTIONS request on any URI that supports use of the extended MKCOL method.

3.1.1 Example: Using OPTIONS for the Discovery of Support for Extended MKCOL

>> Request <<

```
OPTIONS /addressbooks/users/ HTTP/1.1
Host: addressbook.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Allow: OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, COPY, MOVE
Allow: MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, REPORT, ACL
DAV: 1, 2, 3, access-control, extended-mkcol
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Length: 0
```

3.2 Status Codes

As per Section 9.3.1 of [RFC4918].

3.3 Additional Precondition for Extended MKCOL

WebDAV ([RFC4918], Section 16) defines preconditions and postconditions for request behavior. This specification adds the following precondition for the extended MKCOL request.

Name: valid-resourcetype
 Namespace: DAV:
 Use with: Typically 403 (Forbidden)
 Purpose: (precondition) -- The server MUST support the specified resourcetype value for the specified collection.

3.4 Example: Successful Extended MKCOL Request

This example shows how the extended MKCOL request is used to create a collection of a fictitious type "special-resource". The response body is empty as the request completed successfully.

>> Request <<

```
MKCOL /home/special/ HTTP/1.1
Host: special.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol xmlns:D="DAV:" xmlns:E="http://example.com/ns/">
  <D:set>
    <D:prop>
      <D:resourcetype>
        <D:collection/>
        <E:special-resource/>
      </D:resourcetype>
      <D:displayname>Special Resource</D:displayname>
    </D:prop>
  </D:set>
</D:mkcol>
```

>> Response <<

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
```

3.5 Example: Unsuccessful Extended MKCOL Request

This example shows an attempt to use the extended MKCOL request to create a collection of a fictitious type "special-resource", which is not actually supported by the server. The response body shows that an error occurred specifically with the DAV:resourcetype property.

>> Request <<

```

MKCOL /home/special/ HTTP/1.1
Host: special.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol xmlns:D="DAV:" xmlns:E="http://example.com/ns/">
  <D:set>
    <D:prop>
      <D:resourcetype>
        <D:collection/>
        <E:special-resource/>
      </D:resourcetype>
      <D:displayname>Special Resource</D:displayname>
    </D:prop>
  </D:set>
</D:mkcol>

```

>> Response <<

```

HTTP/1.1 403 Forbidden
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol-response xmlns:D="DAV:">
  <D:propstat>
    <D:prop>
      <D:resourcetype/>
    </D:prop>
    <D:status>HTTP/1.1 403 Forbidden</D:status>
    <D:error><D:valid-resourcetype /></D:error>
    <D:responsedescription>Resource type is not
      supported by this server</D:responsedescription>
  </D:propstat>
  <D:propstat>
    <D:prop>
      <D:displayname/>
    </D:prop>
    <D:status>HTTP/1.1 424 Failed Dependency</D:status>
  </D:propstat>
</D:mkcol-response>

```

4. Using Extended MKCOL as an Alternative for MKxxx Methods

One of the goals of this extension is to eliminate the need for other extensions to define their own variant of MKCOL to create the special collections they need. This extension can be used as an alternative to existing MKxxx methods in other extensions as detailed below. If a server supports this extension and the other extension listed, then the server MUST support use of the extended MKCOL method to achieve the same result as the MKxxx method of the other extension.

4.1 MKCALENDAR Alternative

CalDAV defines the MKCALENDAR method to create a calendar collection as well as to set properties during creation (Section 5.3.1 of [RFC4791]).

The extended MKCOL method can be used instead by specifying both DAV:collection and CALDAV:calendar-collection XML elements in the DAV:resourcetype property, set during the extended MKCOL request.

4.1.1 Example: Using MKCOL Instead of MKCALENDAR

The first example below shows an MKCALENDAR request containing a CALDAV:mkcalendar XML element in the request body and returning a CALDAV:mkcalendar-response XML element in the response body.

>> MKCALENDAR Request <<

```
MKCALENDAR /home/lisa/calendars/events/ HTTP/1.1
Host: calendar.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:mkcalendar xmlns:D="DAV:" xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:set>
    <D:prop>
      <D:displayname>Lisa's Events</D:displayname>
    </D:prop>
  </D:set>
</C:mkcalendar>
```

>> MKCALENDAR Response <<

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:mkcalendar-response xmlns:D="DAV:" 
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:propstat>
        <D:prop>
            <D:displayname/>
        </D:prop>
        <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
</C:mkcalendar-response>
```

The second example shows the equivalent extended MKCOL request with the same request and response XML elements.

>> MKCOL Request <<

```
MKCOL /home/lisa/calendars/events/ HTTP/1.1
Host: calendar.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol xmlns:D="DAV:" 
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:set>
        <D:prop>
            <D:resourcetype>
                <D:collection/>
                <C:calendar/>
            </D:resourcetype>
            <D:displayname>Lisa's Events</D:displayname>
        </D:prop>
    </D:set>
</D:mkcol>
```

>> MKCOL Response <<

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol-response xmlns:D="DAV:" 
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:propstat>
        <D:prop>
            <D:resourcetype/>
            <D:displayname/>
        </D:prop>
        <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
</D:mkcol-response>
```

5. XML Element Definitions

5.1 mkcol XML Element

Name: mkcol

Namespace: DAV:

Purpose: Used in a request to specify properties to be set in an extended MKCOL request, as well as any additional information needed when creating the resource.

Description: This XML element is a container for the information required to modify the properties on a collection resource as it is created in an extended MKCOL request.

Definition: `<!ELEMENT mkcol (set+)>`

5.2 mkcol-response XML Element

Name: mkcol-response

Namespace: DAV:

Purpose: Used in a response to indicate the status of properties that were set or failed to be set during an extended MKCOL request.

Description: This XML element is a container for the information returned about a resource that has been created in an extended MKCOL request.

Definition: `<!ELEMENT mkcol-response (propstat+)>`

6. Security Considerations

This extension does not introduce any new security concerns beyond those already described in HTTP [RFC2616] and WebDAV [RFC4918].

7. Acknowledgments

Thanks to Bernard Desrusseaux, Mike Douglass, Alexey Melnikov, Julian Reschke, and Simon Vaillancourt.

8. Normative References

[RFC2119] Bradner, S., "[Key words for use in RFCs to Indicate Requirement Levels](#)", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "[Hypertext Transfer Protocol -- HTTP/1.1](#)", RFC 2616, June 1999.

[RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "[Calendaring Extensions to WebDAV \(CalDAV\)](#)", RFC 4791, March 2007.

[RFC4918] Dusseault, L., "[HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)](#)", RFC 4918, June 2007.

[W3C.REC-xml-20081126] Maler, E., Yergeau, F., Paoli, J., Bray, T., and C. Sperberg-McQueen, "[Extensible Markup Language \(XML\) 1.0 \(Fifth Edition\)](#)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126>>.

Author's Address

Cyrus Daboo

Apple Inc.

1 Infinite Loop

Cupertino , CA 95014

USA

EMail: cyrus@daboo.name

URI: <http://www.apple.com/>